UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/733,734	12/11/2003	Giora Biran	FIS920030290US1	8413

45094 7590 04/15/2009

HOFFMAN WARNICK LLC
75 STATE ST
14TH FL
ALBANY, NY 12207

EXAMINER	
MUSA, ABDELNABI O	
ART UNIT	PAPER NUMBER
2446	

NOTIFICATION DATE	DELIVERY MODE
04/15/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

efiplaw@us.ibm.com
PTOCommunications@hoffmanwarnick.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/733,734
Filing Date: December 11, 2003
Appellant(s): BIRAN ET AL.

Carl F. Ruoff
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/19/2008 appealing from the Office action mailed 05/05/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

7,124,205	Craft et al.	10-2006
20004/0064590	Starr et al.	04-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim(s) 1-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Craft et al. Patent No. (US. 7,124,205 B2) and in view of Starr et al Pub. No. (US 2004/0064590 A1)

As per **claim 1**, Craft et al teach a method of handling a data transfer in a network interface controller (NIC) (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60) the method comprising the steps of:

a) receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type (first packet includes first data and second packet includes a second data Col. 37, Line 27; Col. 37, Line 44; Col. 44, Line 10; FIGs. 5, 17. 26) ;

b) calculating a cyclical redundancy check (CRC) for the data transfer (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61), wherein the CRC is one of valid and invalid (the NIC validate the packet Col. 8, Line 10; FIG. 3); and

c) based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer, conducting one of:

1) dropping the data transfer and not confirming reception (dropping the received data Col. 23, Line 8; Col. 40, Line 50);

2) placing the data transfer to a reassembly buffer of the NIC (the NIC queues the packets in a reassembly buffer Col. 22, Line 67); and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer (frame buffers for receiving and transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4-49; Col 14, Line 17; FIGs. 1-3)

Craft et al. does not teach the specifics on the CRC that is a one of a valid and invalid check wherein handling the data transfer based on the validity of the CRC of a DDP segment as calculated in the in the CRC also does not teach the *specifics* on the comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length.

However, Starr et al teach the message packet sent to the host undergoes CRC cyclical redundancy checking then sent across I/O bus and stored in the host memory also teaches comparators that compare data with corresponding signal (i.e. valid signal) to match memory information to output valid information ([0009] [0066] [0130] FIG.16)

It would have been obvious to a person having ordinary skilled in the art at the time the invention was mad to have modified Craft by the teaching of Starr. Because a

cyclical redundancy check (CRC) is intended for error calculation and is based on data type segments whether TCP length segment or MPA length, the valid data segment is then compared to other packet segments in order to output the correct information to the user.

As per **claim 2**, Craft et al. teach the method of claim 1 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60) , wherein step c), 2) (the NIC queues the packets in a reassembly buffer Col. 22, Line 67) is conducted in the case that the data transfer is of the first type (first packet response is used to identify the data transferred Col. 40, Line 3-26; Col 39, Line 49; FIGs. 9, 11, 17, 25).

As per **claim 3**, Craft et al. teach the method of claim 1 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), further comprising the step of determining whether the data transfer includes a single or multiple direct data placement (DDP) segments (the NIC performs determination of message type before transferring the data Col. Col. 17, Line 17; FIG. 3).

As per **claim 4**, Craft et al. teaches the method of claim 3 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein step c), 3) (frame buffers for receiving and transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4,49; Col 14, Line 17; FIGs. 1-3) is conducted in

the case that the data transfer includes multiple DDP segments (NIC processes the multiple packets and multiple TCP, IP Col. 18, Line 7) and all DDP segments have a valid CRC (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61) that is fully contained in a TCP segment (the TCP headers are validated before processing Col 16, Line 16; FIG. 3).

As per **claim 5**, Craft et al. teaches the method of claim 3 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted in the case that the data transfer includes multiple DDP segments (NIC processes the multiple packets and multiple TCP, IP Col. 18, Line 7), a first DDP segment has an invalid CRC (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61), and a DDP header of the first DDP segment is referred by an MPA length associated with a previous DDP segment (the TCP headers length are validated before processing Col 16, Line 16; FIG. 3).

As per **claim 6**, Craft et al. teaches the method of claim 5 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein, in the case that the data transfer includes multiple DDP segments (NIC processes the multiple packets and multiple TCP, IP Col. 18, Line 7), a first DDP segment has an invalid CRC transfer (the header packets are processed and undergo cyclical redundancy checking in the NIC Col. 2, Line 61), and the DDP header of the

first DDP segment is not referred by the MPA length associated with the previous DDP segment (the network sequencer validates the header length received and checksums the header Col. 15, Line 51; FIGs. 10, 11, 25):

step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted in the case that the DDP header is referred by an MPA marker (TCP headers contains flags for reset and fin that may cause the processor Col. 16, Line 17; FIG. 11); and

step c), 2) (the NIC queues the packets in a reassembly buffer Col. 22, Line 67) is conducted in the case that the DDP header is not referred by the MPA marker (the network sequencer validates the header length received and checksums the header Col. 15, Line 51; FIGs. 10, 11, 25).

As per **claim 7**, Craft et al. teaches the method of claim 3 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted in the case that the data transfer includes multiple DDP segments (NIC processes the multiple packets and multiple TCP, IP Col. 18, Line 7) and a last DDP segment extends outside of the TCP segment boundary (adapter that have the ability to process several types of protocols over TCP Col. 14, Line 40; FIG. 9); and

step c), 2) (the NIC queues the packets in a reassembly buffer Col. 22, Line 67) is conducted in the case that the data transfer includes multiple DDP segments (NIC processes the multiple packets and multiple TCP, IP Col. 18, Line 7) and a last DDP

segment does not extend outside of the TCP segment boundary (adapter that have the ability to process several types of protocols over TCP Col. 14, Line 40; FIG. 9).

As per **claim 8**, Craft et al. teaches the method of claim 2 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein step c), 2) (the NIC queues the packets in a reassembly buffer Col. 22, Line 67) is conducted in the case that the data transfer includes a single DDP segment (data maybe transferred from one NIC to another which involves a single transfer Col. 13, Line 10; FIG. 6) and an MPA length associated with the single DDP segment (TCP headers contains flags for reset and fin that may cause the processor Col. 16, Line 17; FIG. 11) is greater than a transmission control protocol (TCP) segment length of the data transfer (the network sequencer validates the header length received and checksums the header Col. 15, Line 51; FIGs. 10, 11, 25).

As per **claim 9**, Craft et al. teaches the method of claim 2 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein step c), 3) (frame buffers for receiving and transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4,49; Col 14, Line 17; FIGs. 1-3) is conducted in the case that the data transfer includes a single DDP segment (data maybe transferred from one NIC to another which involves a single transfer Col. 13, Line 10; FIG. 6) that has: an MPA length associated therewith that equals a TCP segment length (the network sequencer validates the header length received and checksums the header

Col. 15, Line 51; FIGs. 10, 11, 25) and a valid CRC (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61).

As per **claim 10**, Craft et al. teaches the method of claim 2 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted in the case that the data transfer includes a single DDP segment (data maybe transferred from one NIC to another which involves a single transfer Col. 13, Line 10; FIG. 6) that has: an MPA length associated therewith that equals a TCP segment length (the network sequencer validates the header length received and checksums the header Col. 15, Line 51; FIGs. 10, 11, 25), an invalid CRC and a DDP header (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61) that is referred by an MPA length associated with a previous DDP segment (the NIC performs determination of message type before transferring the data Col. Col. 17, Line 17; FIG. 3).

As per **claim 11**, Craft et al. teaches the method of claim 2 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein in the case that the data transfer includes a single DDP segment (data maybe transferred from one NIC to another which involves a single transfer Col. 13, Line 10; FIG. 6) that has: an MPA length associated therewith that equals a TCP segment length (the network sequencer validates the header length received and

checksums the header Col. 15, Line 51; FIGs. 10, 11, 25), an invalid CRC and a DDP header that is not referred by an MPA length associated with a previous DDP segment (the header packets are processed and undergo cyclical redundancy checking in the NIC Col. 2, Line 61):

step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted in the case that the DDP header is referred by an MPA marker (TCP headers contains flags for reset and fin that may cause the processor Col. 16, Line 17; FIG. 11); and

step c), 2) (the NIC queues the packets in a reassembly buffer Col. 22, Line 67) is conducted in the case that the DDP header is not referred by an MPA marker (the network sequencer validates the header length received and checksums the header Col. 15, Line 51; FIGs. 10, 11, 25).

As per **claim 12**, Craft et al. teaches the method of claim 1 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), further comprising the step of setting the data transfer type to the first type when step c), 2) is conducted (first packet response is used to identify the data transferred Col. 40, Line 3-26; Col 39, Line 49; FIGs. 9, 11, 17, 25).

As per **claim 13**, Craft et al. teaches the method of claim 1 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein in the case that step c), 3) (frame buffers for receiving and

transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4,49; Col 14, Line 17; FIGs. 1-3) is conducted on an out-of-order data transfer (the processor in the NIC checks for fragmented or out of order packets Col. 11, Line 3; Col. 22, Line 66) , the method further comprises the steps of:

clearing TCP hole information created by the out-of-order data transfer in a connection context (packet processing sequencer clears bits from the summary queue Col. 35, Line 45; FIG 25); and

stopping receipt reporting for the out-of-order data transfer (protocol management that control the NIC access to the network and receipt of packets Col. 14, Line 66; Col. 14, Line 35; FIGs 1, 10).

As per **claim 14**, Craft et al. teaches the method of claim 1 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein the data transfer includes DDP segments (the NIC performs determination of message type before transferring the data Col. Col. 17, Line 17; FIG. 3), and the calculating step includes calculating a CRC for all DDP segments of the data transfer together (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61).

As per **claim 15**, Craft et al. teaches the method of claim 14 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20,

Line 9-60), wherein the data transfer does not contain an MPA marker (processing data transfer, Abstract; FIGs. 3-4, 28).

As per **claim 16**, Craft et al. teaches the method of claim 14 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), further comprising the steps of: storing a number of retransmission attempts for each data transfer including an error; and storing a largest sequence number (file server that stores and retrieves files Col. 1, Line 58; Col. 6, Line 41; Col. 20, Line 27 FIG. 3).

As per **claim 17**, Craft et al. teaches the method of claim 16 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein in the case that CRC is invalid for the data transfer (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61), which indicates the data transfer is a newly received error-including data transfer (the packet control sequencer includes all information and any errors or data overflow in the buffer Col. 16, Line 47; FIGs. 10-11):

step c), 2) (the NIC queues the packets in a reassembly buffer Col. 22, Line 67) is conducted on the newly received error-including data transfer in the case that the number of retransmission attempts exceeds a maximum retransmission attempt number for that data transfer (the packet control sequencer for error processing before transmitting or storing in buffer Col. 16, Line 47; FIGs. 10-11), and

step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted on the newly received error-including data transfer in the case that the number of retransmission attempts does not exceed a maximum retransmission attempt number for that data transfer (the packet control sequencer for error processing before transmitting or storing in buffer Col. 16, Line 47; FIGs. 10-11); and

wherein in the case that step c), 1) (dropping the received data Col. 23, Line 8; Col. 40, Line 50) is conducted, the method further comprises the steps of:

increasing the number of retransmission attempts for the newly received error-including data transfer by one (error control in each phase for error handling Col. 28, Line 7, 29; Col. 40, Line 61) that ; and

updating the largest sequence number to carry the largest sequence number among at least one previously received error-including data transfer and the newly received error-including data transfer (error control in each phase for error handling Col. 28, Line 7, 29; Col. 40, Line 61).

As per **claim 18**, Craft et al. teaches the method of claim 16 (a network interface device for data transfer in a network, Abstract, Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60), wherein in the case that CRC is valid for an in-order data transfer (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61):

a) in the case that a sequence number of the in-order data transfer is greater than the stored largest sequence number (the communication control block 'CCB' maintains state information such as number of messages and order of packets that

have been processed Col. 9, Line 13; Col. 11, Line 4; FIGs. 2-4), the number of retransmission attempts is reset and step c), 3) is conducted (frame buffers for receiving and transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4,49; Col 14, Line 17; FIGs. 1-3); and

b) in the case that the sequence number of the in-order data transfer is not greater than the stored largest sequence number (the communication control block 'CCB' maintains state information such as number of messages and order of packets that have been processed Col. 9, Line 13; Col. 11, Line 4; FIGs. 2-4), step c), 3) is conducted (frame buffers for receiving and transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4,49; Col 14, Line 17; FIGs. 1-3).

Regarding claims 19-36 are related to the same limitation set for hereinabove, where the difference used is the phrase 'method' is used hereinabove in the claims, the citations from the prior art has been inserted where's necessary. Furthermore, the wordings of the claims were interchanged within the claim itself and this change does not effect the limitation of the above treated claims. The claim's limitations seemed to be repeated in many claims throughout the application. Even in the above treated claims many of the statements were repeated from previously written claims within the application. Even though claims 19-36 have been differently written from the above treated claims, yet the limitations did not change. As mentioned, claim 19 is the same as claim 1 where the only difference is 'storage means' that was explained in claim 16 whereas claim 20 is the same as claim 2, claim 21 is the same as claim 3, claim 22 is

the same as claim 4, claim 23 is the same as claim 5, claim 24 is the same as claim 6, claim 25 is the same as claim 7, claim 26 is the same as claim 8, claim 27 is the same as claim 9, claim 28 is the same as claim 10, claim 29 is the same as claim 11, claim 30 is the same as claim 12, claim 31 is the same as claim 13, claim 32 is the same as claim 14, claim 33 is the same as claim 15, claim 34 is the same as claim 16, claim 35 is the same as claim 17, claim 36 is the same as claim 18.

As per **claim 37**, Craft et al. teach a computer program product comprising a tangible computer useable medium having computer readable program code embodied therein, which, when executed by a computer infrastructure, enables the computer infrastructure to handle a data transfer in a network interface controller (NIC), the program product (a computer program that contain instructions to run applications Col. 4, Line 24- 64; FIG. 27; Col. 38, Line 4) comprising:

program code configured (Col. 32, Line41) to receive the data transfer wherein the data transfer is denoted as one of a first type and a second type (first packet includes first data and second packet includes a second data Col. 37, Line 27; Col. 37, Line 44; Col. 44, Line 10; FIGs. 5, 17. 26);

program code configured t(Col. 32, Line41) o calculate a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid (the packets undergo cyclical redundancy checking in the NIC Col. 2, Line 61),

program code configured (Col. 32, Line41) to conduct, based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and

validity of a CRC of a direct data placement (DDP) segment within the data transfer, one of:

1) dropping the data transfer and not confirming reception (dropping the received data Col. 23, Line 8; Col. 40, Line 50);

2) placing the data transfer to a reassembly buffer of the NIC (the NIC queues the packets in a reassembly buffer Col. 22, Line 67); and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer (frame buffers for receiving and transmitting packets to a network Col. 7, Line 21; Col. 10, Line 4,49; Col 14, Line 17; FIGs. 1-3)

Craft et al. does not teach the specifics on the CRC that is a one of a valid and invalid check wherein handling the data transfer based on the validity of the CRC of a DDP segment as calculated in the in the CRC also does not teach the *specifics* on the comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length. However, Starr et al teach the message packet sent to the host undergoes cyclical redundancy checking then sent across I/O bus and stored in the host memory also teaches comparators that compare data with corresponding signal (i.e. valid signal) to match memory information to output valid information ([0009] [0066] [0130] FIG.16)

It would have been obvious to a person having ordinary skilled in the art at the time the invention was mad to have modified Craft by the teaching of Starr. Because a cyclical redundancy check (CRC) is intended for error calculation and is based on data type segments whether TCP length segment or MPA length, the valid data segment is then compared to other packet segments in order to output the correct information to the user.

As per **claim 38**, Craft et al. teaches the program product of claim 37, further comprising program code configured to set the data transfer type to the first type when the conducting program code conducts c), 2) (first packet response is used to identify the data transferred Col. 40, Line 3-26; Col 39, Line 49; FIGs. 9, 11, 17, 25)

As per **claim 39**, Craft et al. teaches the program product of claim 37, further comprising program code configured to clear TCP hole information in a connection context and stop receipt reporting (packet processing sequencer clears bits from the summary queue Col. 35, Line 45; FIG 25)for an out-of-order data transfer upon which the conducting program code conducts c), 3) (the processor in the NIC checks for fragmented or out of order packets Col. 11, Line 3; Col. 22, Line 66)

As per **claim 40**, Craft et al. teaches the program product of claim 37, wherein the conducting program code conducts c), 2) in the case that the data transfer is of the

first type (first packet response is used to identify the data transferred Col. 40, Line 3-26; Col 39, Line 49; FIGs. 9, 11, 17, 25)

(10) Response to Argument

- **Argument**

Applicant recites that the references do not disclose, teach or suggest "a method of handling a data transfer in a network interface controller (NIC) wherein.... c) based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer, conducting one off ....1) dropping the data transfer and not confirming reception, as presented in the claims"

- **Response to Argument**

In contrary, even though applicant is claming one of the limitations cited in the claims 1-3, yet the cited art teaches a network interface device for data transfer in a network interface controller or an intelligent network interface card (INIC) that provides processing mechanisms for accelerating data transfers between a network and a storage unit (client), the interface device can use a dedicated fast-path for data transfer between the network and the storage unit. The storage unit, which may include a redundant array of independent disks (RAID) or other configurations of multiple drives, may be connected to the interface device by a parallel channel such as SCSI or by a serial channel such as Ethernet, and the interface device may be connected to the local host by an I/O bus such as a PCI bus. As illustrated in FIG.14 For fast-path UDP data transfer from client 602 to server 600, file streams of data from client application 663 or

audio/video interface 677, The application 663 can arrange the file streams to contain about 8 KB for instance, each file stream received by INIC 606 being pre-pended by INIC 606 with a UDP header according to the socket that has been designated, creating UDP data-grams. The UDP data-grams are divided by INIC 606 into six 1.5 KB message fragments that are each pre-pended with IP and MAC layer headers to create IP packets that are transmitted on network 604. (Craft -Col. 1, Line 60; Col. 10, Line 30; Col. 20, Line 9-60) Now, INIC 622 receives the Ethernet frames from network 604 that were sent by INIC 606, checksums and processes the headers to determine the protocols involved, and sends the UDP and upper layer headers to the AUDP layer 655 to obtain a list of destination addresses for the data from the packets of that UDP datagram. After the packet containing the UDP and upper layer headers has been processed to obtain the destination addresses, the queued data can be written to those addresses. To account for the possibility that not all packets from a UDP datagram arrive, the INIC 622 may use a timer that triggers dropping the received data (Craft - Col.22, line 43-Col.23, Line 23) Error conditions can cause, in a third step, processing of the ISCSI read request command by computer 2401 to switch from fast-path processing to slow-path processing. In such case, network layer and transport layer and session layer processing occur on the host computer. In one example, each of the TCP packets received on network interface device 2408 (packets for the connection of the ISCSI read request command) has a sequence number, the packets are expected to be received with sequentially increasing sequence numbers. If a packet is received with a sequence number that is out of sequence, then an error may have occurred. If,

for example, packets numbered 1, 2, 3, and then 5 are received in that order by network interface device 2408, then it is likely that packet number 4 was **dropped** somewhere in the Network (not as applicant arguing that this data packet is flushed back to host computer) however, under certain conditions, including the out-of-sequence situation in this example, network interface device 2408 determines that a condition has occurred that warrants the connection being "flushed" (means error has been occurred Col.41, Line 16)back to the host computer for slow-path processing. Flushing of the connection entails making CCB 2418 on the network interface device 2408 invalid and making the shadow CCB 2417 on the host computer valid not returning the data packets that were lost. Network interface device 2408 stops fast-path processing of packets for the flushed connection and packets for the flushed connection are passed to the protocol stack of the host computer for slow-path processing. Error handling and/or exception handling are therefore done by the protocol stack in software (Craft -Col.40, Line26-63; FIG.27; Col. 23, Line 8)

In addition, the cited teaches a network interface card (NIC) that provides a physical connection between a host and a network, when a network message packet sent to the host arrives at the NIC, MAC layer headers for that packet are processed and the packet undergoes cyclical redundancy checking (CRC) in the NIC. The packet is then sent across an input/output (I/O) bus such as a peripheral component interconnect (PCI) bus to the host, and stored in host memory. After all the header layers for that packet have been processed, the payload data from the packet is grouped in a file cache with other similarly-processed payload packets of the message.

The data is reassembled by the CPU according to the file system as file blocks for storage on a disk. After all the packets have been processed and the message has been reassembled as file blocks in the file cache, the file is sent as illustrated in FIG.16 and 18 (Starr -[0009] [0066] [0130] FIG.16)

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/A. O. M./

Examiner, Art Unit 2446

April 06, 2009

/Jeffrey Pwu/

Supervisory Patent Examiner, Art Unit 2446

Conferees:

/Jeffrey Pwu/

Supervisory Patent Examiner, Art Unit 2446

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451

***